

Logical Conditions and Booleans



Orbit Görev İçin Hazır mı? 🗸 🗙





Senaryo

Orbit'in yeni bir göreve çıkabilmesi için bazı şartlar var:

- Batarya seviyesi %50'den büyük olmalı
- Sıcaklık 30 dereceden küçük olmalı
- ❖ Ve tabii ki sistem hatasız olmalı!

Orbit bu koşulları kontrol edecek. Eğer hepsi uygunsa görev başlasın! 🚀



- ★ Batarya seviyesi, sıcaklık ve sistem durumu için değişkenler oluştur.
- ★>, <, ==, !=, and, or, not gibi operatörlerle kontrol yap.
- ★ Sonuçları ekrana yazdır.



Logical Conditions and Booleans

Bilmen Gerekenler:

📊 Karşılaştırma Operatörleri:

- $== \rightarrow eşittir$
- $!= \rightarrow$ eşit değildir
- > → büyüktür
- < → küçüktür
- >= \rightarrow büyük veya eşit
- <= → küçük veya eşit

Mantıksal Operatörler:

- and \rightarrow ve (hepsi doğruysa True)
- or \rightarrow veya (biri bile doğruysa True)
- not → değil (True'yi False yapar, tersini alır)

- Karşılaştırma ve mantıksal operatörleri öğrendim!
- ✓ Birden fazla durumu kontrol etmeyi başardım!
- ✓ Orbit'in görev için hazır olup olmadığını anlayabiliyorum! m







Orbit Görev İsimlerini Kıyaslıyor! 🔀 🔍





Senaryo

Orbit'in iki farklı görev planı var: "Mars_Gorevi" ve "Ay_Gorevi" 🞾

Görev sırasını alfabetik olarak kontrol etmesi gerekiyor.

Python'da string değerleri de karşılaştırabiliriz! Nasıl mı? 👇



Görev

- ★ İki farklı string (yazı) değişkeni tanımla.
- ★ Bu değişkenleri karşılaştır (>, <, ==, !=).
 </p>
- ★ Sonuçları ekrana yazdır!



Bilmen Gerekenler:

- 1- Python yazıları da karşılaştırabilir! 🤓
- 2- $== \rightarrow$ Aynı mı diye kontrol eder.
- 3- $!= \rightarrow$ Farklı mı diye kontrol eder
- 4- < ve > → Yazıları **alfabetik sıraya** göre karşılaştırır.

Örneğin: "Ay" yazısı "Mars" yazısından önce gelir, çünkü A harfi M harfinden önce gelir! 🦙





```
#  orbit'in görev adlar1
gorev1 = "Mars_Gorevi"

gorev2 = "Ay_Gorevi"

#   Görevleri karşılaştıralım!
print("Görevler aynı mı?   ", gorev1 == gorev2)
print("Görevler farklı mı?   ", gorev1 != gorev2)
print("Hangisi önce gelir?   ", gorev1 < gorev2)
print("Hangisi sonra gelir?  ", gorev1 > gorev2)
```

- ✓ String (yazı) türündeki değişkenlerle çalıştım!
- ✓ Yazıları karşılaştırmayı öğrendim!
- ✓ Orbit artık görevleri sıraya koyabiliyor! 🕮 ✓



Logical Conditions and Booleans



Orbit Göreve Hazır mı? 🖭





Senaryo

Orbit bir görev için hazırlanıyor! 🚞



Ama önce bir kontrol listesi var:

- ✓ Pili dolu mu?
- ✓ Yazılımı güncel mi?

Eğer ikisi de "evet"se, görev başlasın! 🚀

Yoksa Orbit dinlenmeye çekiliyor...



- 🖈 İki tane mantıksal değişken tanımla: pili_dolu ve guncel_mi
- ★ Eğer her ikisi de doğruysa göreve başla!
- ★ Değilse "Hazır değil" yaz!



Bilmen Gerekenler:

- True → Doğru anlamına gelir
 ✓
- False → Yanlış anlamına gelir X
- and \rightarrow İki şey de doğruysa **True** olur.
- Örnek: Pili dolu ve yazılım güncel → Hazır
- or \rightarrow İkisi veya biri doğruysa **True** olur.
- not → Tersini söyler: not True → False

```
pili_dolu = True
guncel_mi = True

# Göreve hazır mı?
hazir_mi = pili_dolu and guncel_mi

# Sonucu ekrana yazdır
print("Orbit göreve hazır mı? %", hazir_mi)
```

*

- Boolean değişkenlerle çalışmayı öğrendim!
- ✓ and ile iki şartı aynı anda kontrol ettim!
- ✓ Orbit artık görev öncesi kontrol yapabiliyor! 🛠



Logical Conditions and Booleans



Orbit Koşulları Kontrol Ediyor!





Senaryo

Orbit'in programına bazı kurallar yazdık!

Bugün hangi gün olduğunu, boyunun yeterli olup olmadığını ve canının bitip bitmediğini kontrol ediyor.

Bu kontrollerin sonucunda doğruysa True, yanlışsa False yazacak.



Görev

★ 3 koşul oluştur:

- Bugün günlerden Cumartesi mi?
- Orbit'in boyu yeterli mi?
- Orbit'in canı bitti mi?
- Her koşulun sonucunu print() ile yazdır.

Logical Conditions and Booleans

```
# 译 Eğer bugün Cumartesi ise, film izleyebilir
bugun = "Cumartesi"
film_izle = (bugun == "Cumartesi")
print("译 Film izleyebilir mi?", film_izle)

# 术 Eğer boy 48 inç veya daha fazlaysa, rollercoaster'a binebilir
boy = 50
rollercoaster_izni = (boy >= 48)
print("术 Rollercoaster'a binebilir mi?", rollercoaster_izni)

# 茶 Eğer can sayısı 1'den azsa, oyun biter
can_sayisi = 0
oyun_bitti_mi = (can_sayisi < 1)
print("茶 Oyun bitti mi?", oyun_bitti_mi )
```

- Karşılaştırma operatörleri ile koşul yazmayı öğrendim!
- ✓ ==, >=, < gibi işaretlerin ne işe yaradığını gördüm!
- ✓ Koşul doğruysa True, yanlışsa False çıktığını öğrendim!
- ✓ Orbit artık karar verebiliyor! @



Logical Conditions and Booleans



Orbit Bugün Dışarı Çıkabilir mi?





Senaryo

Bugün hava çok güzel! 🔆

Orbit dışarı çıkmak istiyor ama bir robot olarak kurallara uymalı 🛑

Önce kendisine şu iki soruyu sormalı:

- 1. Hava güzel mi?
- 2. Orbit dışarı çıkmak için izinli mi?

Eğer ikisi birden doğruysa pikniğe gidebilir!

Amaaaa... sadece biri doğruysa bile parka gidebilir 🎠

Haydi bu durumu Python ile kontrol edelim! @__



- ★ 2 farklı koşul tanımla (hava_guzel, orbit_izinli)
- 🖈 and ve or kullanarak karar ver
- 🖈 Sonuçları print() ile yazdır

Bilmen Gerekenler:

 \checkmark and = VE

İki koşul da doğruysa sonuç True olur.

Örn: Hava güzel VE izinliyse → Piknik zamanı! ♥

✓ or = VEYA

Koşullardan biri bile doğruysa sonuç True olur.

Örn: Hava güzel VEYA izinli → Parka gidebiliriz! \[
\]
\[
\]

✓ AND (ve) Operatörü Tablosu

← Sadece her iki koşul da doğruysa sonuç True olur.

Şart 1	Şart 2	Sonuç (AND)
TRUE	TRUE	✓ True
TRUE	FALSE	× False
FALSE	TRUE	× False
FALSE	FALSE	× False

OR (veya) Operatörü Tablosu

F Herhangi bir şart doğruysa sonuç True olur.

Şart 1	Şart 2	Sonuç (OR)
TRUE	TRUE	✓ True
TRUE	FALSE	✓ True
FALSE	TRUE	✓ True
FALSE	FALSE	× False



Logical Conditions and Booleans



Orbit Bugün Dışarı Çıkabilir mi?





Senaryo

Bugün hava çok güzel! 🔆

Orbit dışarı çıkmak istiyor ama bir robot olarak kurallara uymalı 🛑

Önce kendisine şu iki soruyu sormalı:

- 1. Hava güzel mi?
- 2. Orbit dışarı çıkmak için izinli mi?

Eğer ikisi birden doğruysa pikniğe gidebilir!

Amaaaa... sadece biri doğruysa bile parka gidebilir 🎠

Haydi bu durumu Python ile kontrol edelim! @__



- ★ 2 farklı koşul tanımla (hava_guzel, orbit_izinli)
- 🖈 and ve or kullanarak karar ver
- 🖈 Sonuçları print() ile yazdır